Introduction

A recent NASA invention enables considerable energy savings when using the common induction electric motor. It can be used with existing motors, since it requires no modification to the motor. Typical energy savings of 10 to 60% can be realized, depending on the amount of motor loading present. This invention is the direct result of an analysis, by NASA, of Solar Heating and Cooling Systems to reduce the power consumed by pump and fan motors used in these systems. It is applicable to both single phase and 3 phase motors.

Since the induction motor is widely used in many of the same systems that can take advantage of the 8022 microcontroller, this invention can provide a significant benefit to many commercial and consumer products. Examples include the following:

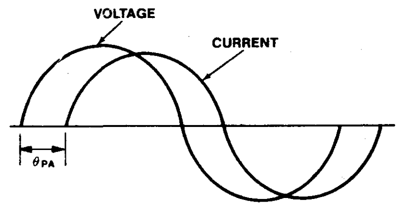
- —space heating and cooling systems
- —heat pumps
- —solar collector controllers
- —liquid or chemical process control
- —large industrial motor control
- —refrigeration units
- —swimming pool controllers
- —washing machines
- —dishwashers.

This application note will explain how this invention can be implemented with an 8022 microcontroller with a minimum amount of external hardware. The note is organized into the following sections:

1. Power Factor Controller theory of operation.
2. Description of the 8022 microcontroller in this application.
3. Hardware description.
4. Software description and listing.
5. Conclusion.

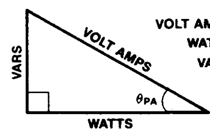
Theory of Operation

The concept of the Power Factor Controller (PFC), conceived and developed by NASA aerospace engineer Frank Nola,[1] is to reduce the voltage applied to the motor when it is partially loaded, resulting in significant energy savings. At full load, the induction motor must have a high flux density in order to perform adequately. Under this condition the motor is running at peak efficiency with a power factor of about 0.8. At less than full load, however, this high flux density is still supported by the high current set up in the field coils. The resulting power factor can drop to as low as 0.1 (FIG. 2). The losses associated with this high current, under the lightly loaded condition, is primarily in the form of heat. This is a loss which can be paid for twice if the motor is in a refrigerated or otherwise cooled system.



θ_{PA} is the phase angle between the voltage and current waveforms (the current will always lag the voltage in an induction motor).

Figure 1.



VOLT AMPS—APPARANT POWER—I_{EFF} V_{EFF}
WATTS— AVERAGE POWER —I_{EFF} V_{EFF} COS θ_{PA}
VARS— REACTIVE POWER —I_{EFF} V_{EFF} SIN θ_{PA}

The Power Factor is equal to the cosine of the phase angle. It represents the ratio of the total power required to be supplied by the power company to the power recorded by a watt meter.

$$\text{Power Factor} = \frac{\text{Average Power}}{\text{Apparent Power}} = \text{Cos}\,\theta_{PA}$$

The Power Factor ranges from 0 to 1. For a purely resistive load, the voltage and current are in phase, θ_{PA} is zero, and the P.F. is unity, the Apparent Power and Average Power are equal.

For a purely reactive load, that is, one containing no resistance, the phase difference between the voltage and current is either plus or minus 90° (purely inductive being plus 90°, purely capacitive being minus 90°), the P.F. is therefore zero, resulting in the Average Power being zero.

Between these two extreme cases, there are the general networks for which the P.F. can range from zero to unity. Therefore, for a constant magnitude of the current and voltage, the Average Power (Watts) can range from zero to a maximum value equal to the Apparent Power, depending on the phase angle.

Figure 2. Complex Power and the Power Triangle

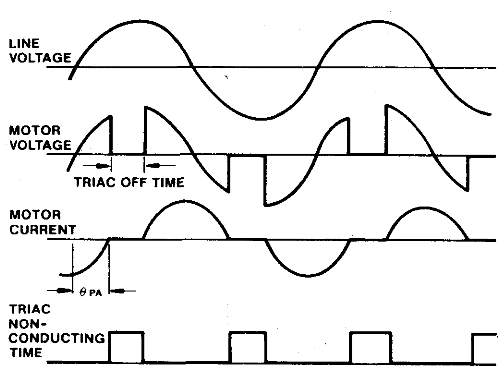


Figure 3.

[1] NASA Tech Brief MFS-23280, Power Factor Controller. Copies may be obtained by writing to: Director, Technology Utilization Office, Marshall Space Flight Center, AL, 35812. The patent for the PFC is owned by the U.S. Government. Licenses for commercial development are available at no charge. Contact: Patent Counsel, Marshall Space Flight Center, AL, 35812.

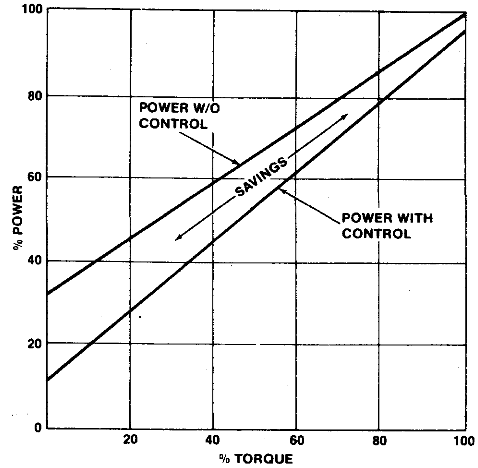
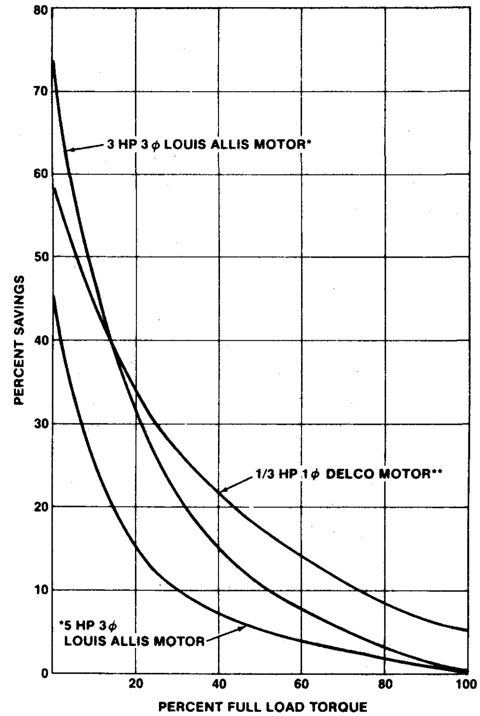


Figure 4. Typical Power Savings[2] for Single Phase Motor



*DATA OBTAINED FROM AUBURN UNIVERSITY REPORT
**NASA/MARSHALL SPACE FLIGHT CENTER (MSFC) DATA

Figure 5. Power Factor Controller Percent Savings Vs. Torque for Various Motors[3]

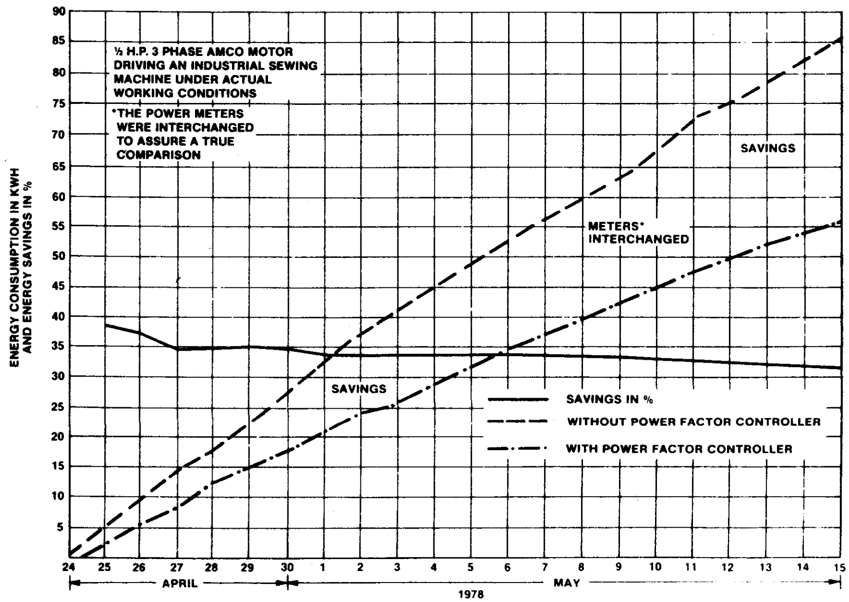


Figure 6. Energy Consumption as a Function of Time[4]

In this partial load condition, the motor would produce the same torque at essentially the same speed even if much weaker magnetic forces were set up, by decreasing the current which produces these fields. The electric motor on its own, can't recognize this condition, however, and will continue to draw near the high current used under full load, even when operating under no load.

The principle of operation of the PFC is to measure the shift in phase angle between motor voltage and current

[2]NASA TECH BRIEF MFS-23280
[3]Ibid.
[4]Ibid.

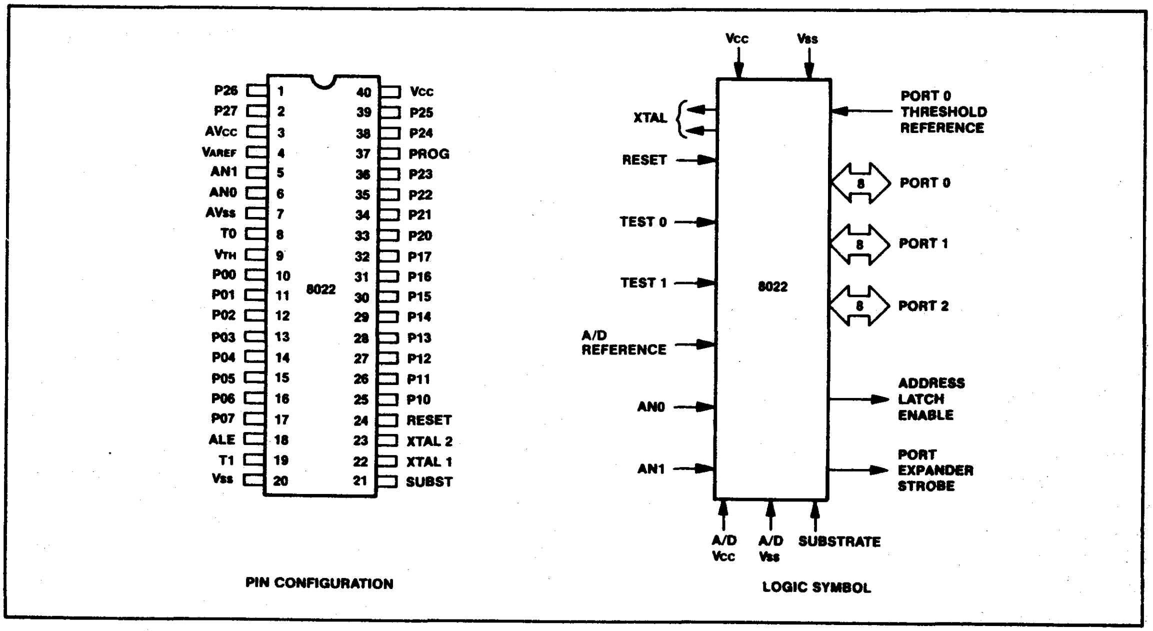


Figure 7.

as the load changes. As the monitored phase angle increases, a solid-state switch (triac) is used to reduce the voltage applied to the motor (FIG. 3). This increases motor slip and reduces the phase angle to a predetermined value. The feedback loop forces the motor to run at a constant, optimum, phase angle selected by the user, thereby minimizing wasted power. Since phase shift can be sampled as often as every power-line cycle, the 8022 can immediately respond to changes in the load and the motor will always run at near-peak efficiency, supplying only the amount of power required.

The PFC was originally tested at the Marshall Space Flight Center on 40 motors ranging from 1/12 hp to 5 hp, both single phase and 3 phase.[5] Most motors exhibited a 40–60% savings at no load and 0–10% savings when fully loaded. The more expensively built 3-phase motors being the more efficient. When the device was used in 15 typical applications with single phase motors, including a drill press and vacuum pump, the typical savings were 30–40% when idling and 10% when loaded. The vacuum pump even ran about 25 degrees C cooler with the device. In a 500 hour test of two identical machines performing the same task, the piece of equipment with the PFC used 33% less energy (FIG. 6).

The device was also tested extensively by the Auburn University and many companies, all coming to the same conclusion. The Power Factor Controller is proven to offer significant savings to the user that, by the way, go beyond the actual direct power reduction mentioned. These multiple savings include the motor running cooler and quieter, extending the motor's life as well as saving in an air conditioned environment since there is less heat generated, and also by actually controlling the power factor to a desired value. This will benefit large users of motors with cyclic loads who are often charged by the Power Company for a poor power factor and are forced to correct this condition with large capacitor banks or a synchronous condenser.

[5] Ibid.

The 8022

The PFC developed by NASA consists of discrete electronic parts, designed to produce an analog controller. The 8022 microcontroller brings the benefit of a programmable computer to this application so that intelligent control of the total system can be effected. The 8022 is especially suited for an analog environment as it contains a 2-channel 8-bit A/D converter, zero-cross detection circuitry, and comparitor inputs with a controllable threshold. All of these capabilities are integrated into a single chip, along with 2048 bytes of program storage, 64 bytes of RAM, an 8-bit internal timer/event counter, external interrupt input, two high current drive outputs, and three 8-bit I/O ports. The result is a complete microcomputer system on a single chip. If the reader desires more information about the 8022, (s)he may refer to the MCS-48 User's Manual and application note AP-56, "Designing With Intel's 8022 Microcontroller," for a complete description of the 8022.

The 8022 is already being used to control many systems which utilize an induction motor. The extra code required for the PFC operation is only 154 bytes (less than 8% of the available program store). The main program would, of course, have to be modified to facilitate the PFC's function. This can easily be performed by placing the PFC operation in an interrupt routine. The total number of pins delegated to the PFC operation is 4: three I/O pins and the T1 zero-cross input. One of these I/O pins is dedicated to control Vth.

These three interface lines: the voltage-cross input, the current-cross input, and the triac gate control, are all that is required for the PFC application. The 8022 simply measures the amount of lag time between the voltage zero-crossing and the current zero-crossing. This measured time is then converted to a phase angle and subsequently compared to the desired phase angle. As the load changes and the measured phase angle shifts either greater than or less than the desired value, the 8022 will either lengthen or shorten the triac off time. The result is that the motor only gets the amount of current needed to drive the instantaneous load.

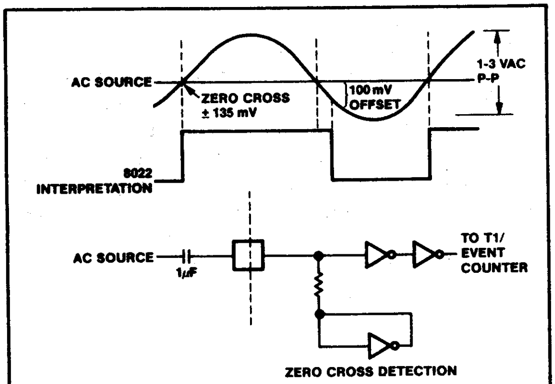


Figure 8.

Hardware Description

To perform the PFC function, the motor was placed in series with a triac which cuts out portions of the applied voltage, and a small series resistor which was used for measuring the current (FIG. 9). To isolate the 8022 from the AC line voltage, a small audio transformer was placed across the current sensing resistor, and an opto-isolator was used between the gate of the triac and the pin driving it. The voltage signal was taken from the secondary side of the power supply transformer.

Aside from the isolation, the only other interface hardware was some simple signal conditioning. Since the T1 zero-cross input requires a 1-3 VAC p-p signal, two diodes to DC ground clipped the AC signal nicely at 1.5 VAC p-p. A resistor for current limiting and a 1 microfarad capacitor to T1 complete the voltage zero-cross.

One of the port 0 comparator inputs was used for the current zero-cross detection. The 8022 had to be able to recognize when the current waveform was approaching zero from both the positive side and the negative side since it doesn't cross zero at any particular point, but rather approaches it from either side then remains zero until the triac fires again (FIG. 3). This was done by having two separate thresholds for the comparator. One for the negative slope crossing and one for the positive. To accomplish this, the current waveform was first boosted up with a DC level so it would be entirely positive. VTH was then biased to this DC level by two resistors. The threshold would be changed by about 0.1v either side of this DC level under software control by writing either a logic "1" or "0" to P17 when anticipating either a positive approach or a negative approach, respectively. This would place the 100k ohm resistor in parallel with either resistor thereby decreasing its value and subsequently raising or lowering the threshold.

The remaining hardware was included to aid in the development of this application note, but is not necessary for the system's operation. The remaining 6 pins on port zero were used to input the desired phase angle. The remaining 7 pins on port 1 were used to control the 7 segments of a display. The upper nibble of port 2 was used to control up to 10 digits of a display. The user would then select the desired phase angle at will, with the actual phase angle being written to the display.

With this pin assignment the remaining pins include the lower nibble of port 2 which can be used with the Intel 8243 I/O expander, adding four 4-bit I/O ports. The T0 pin can still be used as either a testable input, the interrupt request pin, or both. Finally, the two channels of the A/D converter are available, allowing the chip to interface directly with analog transducers. Aside from the Power Factor Controller and the direct user interface with the 8022 (via a keypad or thumbwheel switches and the display), there are still enough pins left over for almost any controller application.

Hardware not required for the PFC operation is shaded in the schematic.

Software Description

To simplify the software, the triac "offtime" is quantized in units of 0.27ms. This represents one time-unit of the timer (operating with a 3.58 MHz TV crystal). The "offtime" is therefore incremented or decremented by one of these units (or remained unchanged) when it is

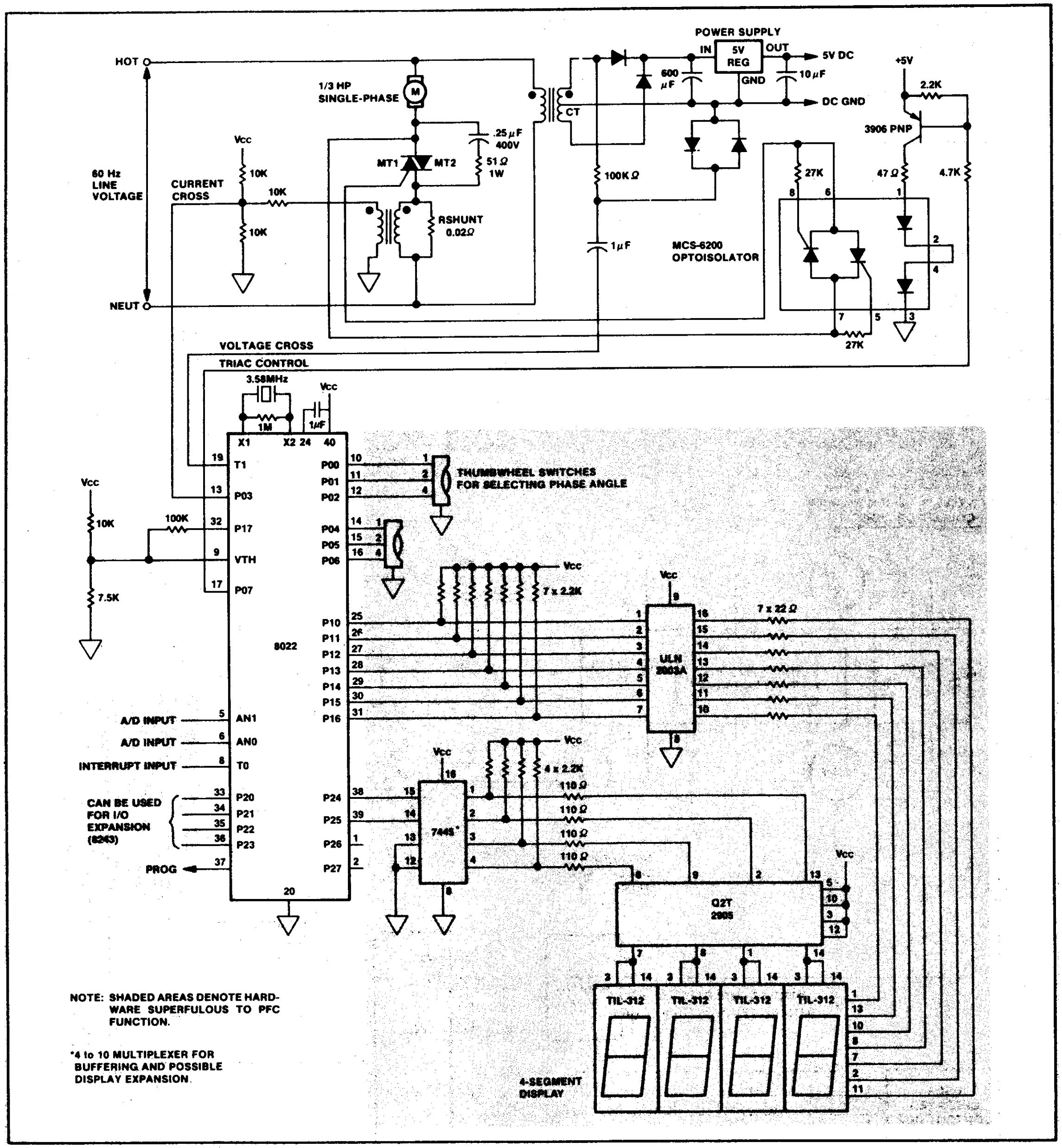


Figure 9.

reviewed for adjustment every cycle. This proves to be an adequate reaction to abrupt changes in the load. If however, the phase angle will only be sampled every 3 cycles, for example, on an interrupt, a simple routine has been included to change the "offtime" by more than one increment should the phase angle change more than 12 degrees between samplings. This is included to compensate for extremely abrupt load changes.

For simplicity, the phase angle is also measured in units of approximately 6 degrees, to coincide with the timer unit of 0.27ms. The lookup table consists of these quantized values. For a steady phase angle, the desired phase angle should be chosen to be one of these values. If any value between two of these are chosen, the phase angle oscillates between these two values and will never equal the chosen value, since this value is not in the lookup

table. This can be rectified by using a table lookup and interpolation instead of a simple table lookup (see AP-24; "Application Techniques for the MCS-48 Family"). There is no apparent benefit in doing this, however, as the value-searching over a 6 degree range doesn't seem to cause any problems.

The software also distinguishes between that necessary for the PFC only, and that used to include the user interface features. The BCD-to-binary routine was adapted from AP-49, "Serial I/O and Math Utilities for the 8049 Microcomputer," as was the binary-to-BCD routine. The total bytes of program storage consumed by the PFC only, is 154, and the total for this entire application is 328.

The software flowchart and complete listing follow.

Conclusion

The advantages of controlling a system with a programable microcontroller are evident. The added advantage of conserving power with the 8022 in the same application is realized in this application note. As the Power Factor Controller will be used in more and more applications as an energy-saving feature, the 8022 is ideally suited to implement it.

To illustrate this, the following application shows the 8022 controlling a Heat Pump / Solar Heating system.

Figure 10 illustrates the type of solar heating system that was the target of the NASA invention. The 8022 can control this system, and in addition, implement the PFC function on the motors driving the heatpump compressor and fan. The A/D can be used to monitor the temperatures in the system, and the various pumps and fans can be activated so as to maximize the heat extracted from the solar radiation. As an example of this, the flow rate in the solar-collector circuit can be adjusted by the 8022 so that heat can be collected even on cloudy days. (Lowering the flow rate lets the fluid remain in the collectors longer so that the heat radiation has a longer period in which to heat the fluid.) The 2K bytes of program storage available on the 8022 would make the addition of features such as a real-time clock for controlling the temperature to different levels at different times of the day, and dual-level thermostat settings, possible at little or no incremental product cost.

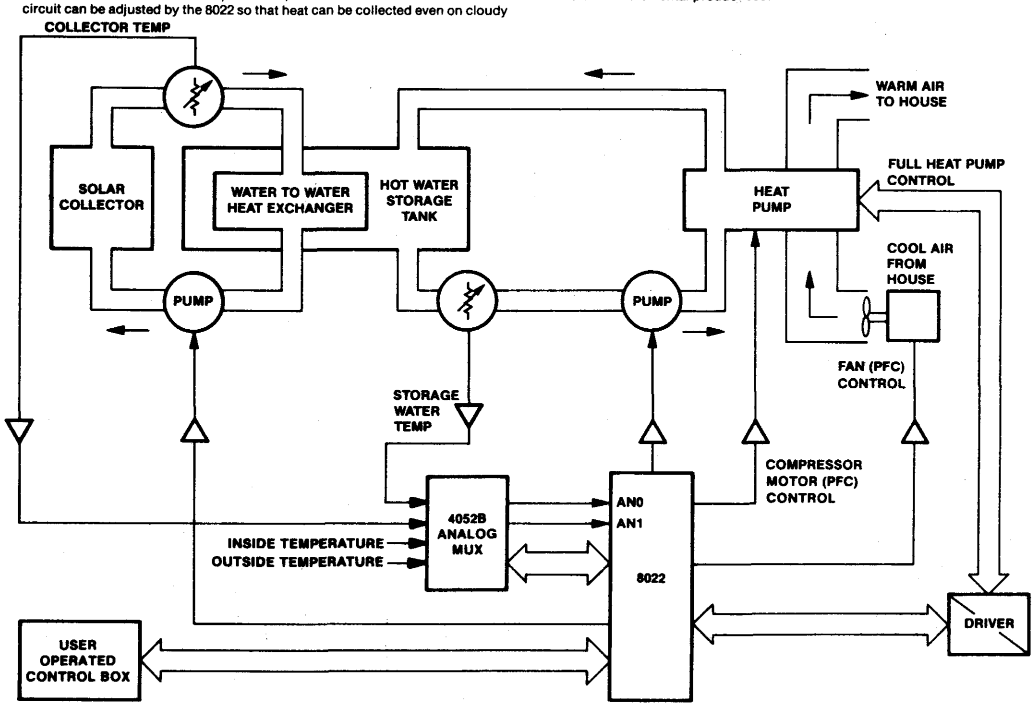
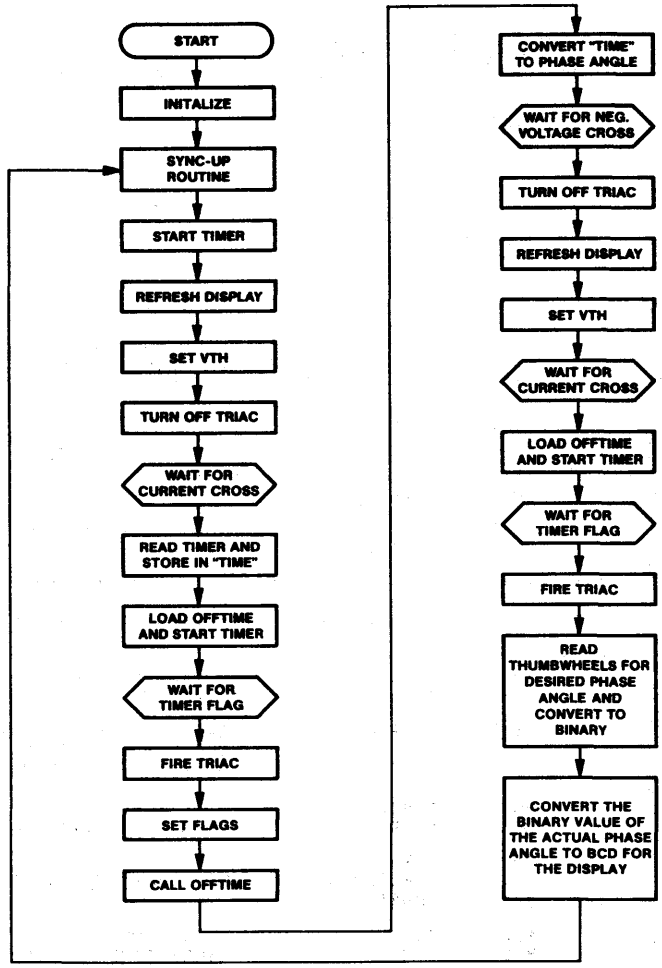


Figure 10.



Program Flow Chart

```
ASM48 AP PAGELENGTH(88)

ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    1

LOC  OBJ          SOURCE STATEMENT

                  $MOD22 SYMBOLS MACROFILE XREF
                  ;**********************************************************************
                  ;        DEFINITIONS OF MACROS
                  ;**********************************************************************
                  LDA     MACRO    PAR1
                          MOV      RO,#PAR1
                          MOV      A,@RO
                          ENDM
                  STA     MACRO    PAR1
                          MOV      RO,#PAR1
                          MOV      @RO,A
                          ENDM
0000 1498         INIT:   CALL     INITZ             ;INITIALIZE
                  ;**********************************************************************
                  ;          THE FOLLOWING IS A SYNC-UP ROUTINE TO ASSURE
                  ;           A START ON A RAISING VOLTAGE ZERO-CROSS
                  ;**********************************************************************
0002 5606         PWRON:  JT1      VOWAIT            ;JUMP ON T1=1
0004 0402                 JMP      PWRON
0006 460A         VOWAIT: JNT1     WAITST            ;JUMP ON T1=0
0008 0406                 JMP      VOWAIT
000A 560E         WAITST: JT1      START             ;JUMP ON T1=1
000C 040A                 JMP      WAITST
                  ;**********************************************************************
                  ;         END OF SYNC-UP             ==START PROGRAM==
                  ;**********************************************************************
000E 27           START:  CLR      A
000F 62                   MOV      T,A               ;CLEAR PRESCALER
0010 55                   STRT     T                 ;START TIMER
0011 3400                 CALL     REFRSH            ;REFRESH THE DISPLAY
                                                     ;VTH SET FOR UPCOMING CURRENT AUTO-
                                                     ;MATICALLY IN REFRSH ROUTINE (P17=0)
0013 23FF                 MOV      A,#0FFH
0015 90                   OUTL     PO,A              ;TURN OFF TRIAC
0016 08           IPWAIT: IN       A,PO
0017 5308                 ANL      A,#00001000B      ;WAIT FOR CURRENT CROSS
0019 C616                 JZ       IPWAIT
001B 42                   MOV      A,T               ;READ TIMER
001C AA                   MOV      TIME,A            ;STORE TIMER VALUE
001D 14BB                 CALL     TRIAC
                  $EJECT
```

NOTE: SHADED AREAS DENOTE SOFTWARE SUPERFULOUS TO PFC FUNCTION.

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    2

LOC  OBJ         SEQ        SOURCE STATEMENT

                     ;*********************************************************************
                     ;                         SET THE FLAGS
                     ;*********************************************************************
001F 97                      CLR     C                   ;CLEAR THE CARRY FLAG
0020 FC                      MOV     A,SETVAL
0021 37                      CPL     A
0022 17                      INC     A                   ;2'S COMPLIMENT SETVAL
0023 6D                      ADD     A,PHSANG            ;PHSANG-SETVAL = DELTA
0024 C64F                    JZ      LOOK                ;IF PHSANG=SETVAL, DONT CHANGE ANYTHING
                             STA     DELTA
0026 B826                    MOV     R0,#DELTA
0028 A0                      MOV     @R0,A
0029 5380                    ANL     A,#10000000B        ;TEST FOR NEG
002B C638                    JZ      OTCALC              ;IF POS DO NOT SET FLAG
002D 1E                      INC     PHLTSV              ;SET PHSANG L.T. SETVAL FLAG
                             LDA     DELTA
002E B826                    MOV     R0,#DELTA
0030 F0                      MOV     A,@R0
0031 030A                    ADD     A,#10D              ;10-DELTA
0033 5380                    ANL     A,#10000000B
0035 C638                    JZ      OTCALC
0037 1F                      INC     DLGT10              ;SET DELTA G.T. 10 FLAG IF NEG
                     ;*********************************************************************
                     ;           CHANGE OFFTIME PER FLAGS NEWLY SET
                     ;*********************************************************************
0038 FE              OTCALC: MOV     A,PHLTSV            ;DECREMENT OR INCREMENT?
0039 C649                    JZ      DECREM              ;IF PHLTSV IS CLEAR THEN DECREMENT
003B FB              INCREM: MOV     A,OFFTIM            ;IF OFFTIME=255 OR 254 DON'T INCREMENT
003C 37                      CPL     A
003D C64C                    JZ      CLRFLG
003F 07                      DEC     A
0040 C64C                    JZ      CLRFLG
0042 1B                      INC     OFFTIM              ;INCREMENT
0043 FF                      MOV     A,DLGT10            ;SHOULD IT INCREMENT TWICE?
0044 C64C                    JZ      CLRFLG
0046 1B                      INC     OFFTIM
0047 044C                    JMP     CLRFLG
0049 FB              DECREM: MOV     A,OFFTIM
004A 07                      DEC     A
004B AB                      MOV     OFFTIM,A            ;DECREMENT
004C 27              CLRFLG: CLR     A
004D AE                      MOV     PHLTSV,A
004E AF                      MOV     DLGT10,A            ;CLEAR FLAGS
                     $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0                PAGE     3

 LOC  OBJ          SEQ         SOURCE STATEMENT

                        ;*******************************************************************
                        ;                LOOKUP VALUE FOR "PHSANG" FROM "TIME"
                        ;*******************************************************************

004F 23C9               LOOK:    MOV     A,#(LOW TAB)     ;BOTTOM OF TABLE
0051 6A                          ADD     A,TIME
0052 A3                          MOVP    A,@A             ;GET PHSANG VALUE
0053 AD                          MOV     PHSANG,A

                        ;*******************************************************************
                        ;                WAIT FOR NEGATIVE GOING VOLTAGE CROSS
                        ;*******************************************************************

0054 4658               HOLD:    JNT1    CONTON
0056 0454                        JMP     HOLD

0058 23FF               CONTON:  MOV     A,#0FFH          ;TURN OFF TRIAC
005A 90                          OUTL    P0,A

005B 3400                        CALL    REFRSH           ;REFRESH THE DISPLAY
                                 LDA     SEGDAT
005D B824                        MOV     R0,#SEGDAT
005F F0                          MOV     A,@R0
0060 4380                        ORL     A,#10000000B     ;SET VTH FOR DOWN GOING CURRENT-
0062 39                          OUTL    P1,A             ;BY SETTING P17

0063 08                 INWAIT:  IN      A,P0             ;WAIT FOR CURRENT CROSS
0064 5308                        ANL     A,#00001000B
0066 9663                        JNZ     INWAIT

0068 14BB                        CALL    TRIAC

                        ;*******************************************************************
                        ;                     READ THUMBWHEELS
                        ;                 AND CONVERT TO BINARY
                        ;*******************************************************************

006A 08                          IN      A,P0
006B 5377                        ANL     A,#01110111B
                                 STA     BINTR            ;STORE ACC IN BINTR
006D B820                        MOV     R0,#BINTR
006F A0                          MOV     @R0,A

0070 14A7                        CALL    SELRB1

                                 LDA     BINTR
0072 B820                        MOV     R0,#BINTR
0074 F0                          MOV     A,@R0

0075 344A                        CALL    CONBIN           ;CONVERT THUMBWHEELS TO BINARY

                                 STA     BINTR            ;STORE ACC IN BINTR
0077 B820                        MOV     R0,#BINTR
0079 A0                          MOV     @R0,A

007A 14B1                        CALL    SELRB0

                                 LDA     BINTR            ;MOV BINTR TO ACC
007C B820                        MOV     R0,#BINTR
007E F0                          MOV     A,@R0
007F AC                          MOV     SETVAL,A

                        $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    4

LOC  OBJ        SEQ        SOURCE STATEMENT

                     ;**********************************************************************
                     ;        CONVERT BINARY VALUE OF THE PHASE ANGLE (PHSANG)
                     ;                      TO BCD (ANGLE)
                     ;**********************************************************************

0080 FD                     MOV     A,PHSANG
                            STA     BCDTR          ;STORE PHASE ANGLE
0081 B821               +   MOV     R0,#BCDTR
0083 A0                 +   MOV     @R0,A

0084 14A7                   CALL    SELRB1

                            LDA     BCDTR
0086 B821               +   MOV     R0,#BCDTR
0088 F0                 +   MOV     A,@R0

0089 3437                   CALL    CNBCD

                            STA     BCDTR
008B B821               +   MOV     R0,#BCDTR
008D A0                 +   MOV     @R0,A

008E 14B1                   CALL    SELRB0

                            LDA     BCDTR
0090 B821               +   MOV     R0,#BCDTR
0092 F0                 +   MOV     A,@R0
                            STA     ANGLE
0093 B823               +   MOV     R0,#ANGLE
0095 A0                 +   MOV     @R0,A


0096 040A                   JMP     WAITST         ;GO BACK TO BEGINNING AND REPEAT

                     ;**********************************************************************
                     ;**********************************************************************
                     ;**********************************************************************
                     ;   ------------------- END OF MAIN PROGRAM ----------------------
                     ;**********************************************************************
                     ;**********************************************************************
                     ;**********************************************************************

                     $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    5

 LOC  OBJ          SEQ        SOURCE STATEMENT

                         INITZ:

                         ;**************************************************************
                         ;                ALLOCATE     RAM
                         ;**************************************************************

0020                     BINTR    EQU      20H
0021                     BCDTR    EQU      21H
0022                     PASTOR   EQU      22H
0023                     ANGLE    EQU      23H
0024                     SEGDAT   EQU      24H
0025                     DISFLG   EQU      25H
0026                     DELTA    EQU      26H

0002                     TIME     EQU      R2
0003                     OFFTIM   EQU      R3
0004                     SETVAL   EQU      R4
0005                     PHSANG   EQU      R5
0006                     PHLTSV   EQU      R6
0007                     DLGT10   EQU      R7

                         ;**************************************************
                         ;        FOR USE IN CONBIN AND CONBCD
                         ;        ****************************

0002                     XA       EQU      R2
0003                     COUNT    EQU      R3
0004                     ICNT     EQU      R4
0001                     DIGPR    EQU      1

                         ;**************************************************

0098 BC29                         MOV      R4,#29H             ;SETVAL=41
009A BD29                         MOV      R5,#29H             ;PHSANG=41
009C BBFF                         MOV      R3,#0FFH            ;SET OFFTIM TO 255

009E 237F                         MOV      A,#7FH              ;TURN ON TRIAC
00A0 90                           OUTL     P0,A

00A1 97                           CLR      C
00A2 27                           CLR      A
                                  STA      DISFLG              ;CLEAR DISPLAY FLAG
00A3 B825                         MOV      R0,#DISFLG
00A5 A0                           MOV      @R0,A

00A6 83                           RET

                         $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    6

 LOC  OBJ          SEQ        SOURCE STATEMENT

                    ;**********************************************************************
                    ;            THESE NEXT TWO SUBROUTINES ACT LIKE A BANK SELECT
                    ;                EXCEPT THAT R0 AND R1 GET WIPED OUT
                    ;**********************************************************************
                    ;
                    SELRB1:
 00A7 B918                  MOV     R1,#18H
 00A9 B807                  MOV     R0,#07H
 00AB F0            BAK1:   MOV     A,@R0
 00AC A1                    MOV     @R1,A
 00AD 19                    INC     R1
 00AE E8AB                  DJNZ    R0,BAK1
 00B0 83                    RET
                    ;**********************************************************************
                    ;
                    SELRB0:
 00B1 B918                  MOV     R1,#18H
 00B3 B807                  MOV     R0,#07H
 00B5 F1            BAK2:   MOV     A,@R1
 00B6 A0                    MOV     @R0,A
 00B7 19                    INC     R1
 00B8 E8B5                  DJNZ    R0,BAK2
 00BA 83                    RET
                    $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    7

 LOC  OBJ          SEQ        SOURCE STATEMENT

                    ;**********************************************************************
                    ;                    THIS ROUTINE FIRES THE TRIAC AFTER A
                    ;                           PREDETERMINED OFFTIME
                    ;**********************************************************************
                    ;
                    TRIAC:
 00BB 65                    STOP    TCNT            ;STOP TIMER
 00BC 16BE                  JTF     $+2             ;CLEAR TIMER FLAG

 00BE FB                    MOV     A,OFFTIM        ;SET PRESCALER TO OFFTIME
 00BF 62                    MOV     T,A

 00C0 55                    STRT    T               ;START TIMER

 00C1 16C5          CNTDWN: JTF     FIRE            ;WAIT FOR TIMER FLAG
 00C3 04C1                  JMP     CNTDWN

 00C5 237F          FIRE:   MOV     A,#7FH          ;FIRE
 00C7 90                    OUTL    P0,A            ;TRIAC

 00C8 83                    RET

 00C9               TAB     EQU     $

                    $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0            PAGE    8

 LOC  OBJ         SEQ        SOURCE STATEMENT

                  337
                  338 ;*****************************************************************
                  339 ;         THIS ROUTINE REFRESHES THE DISPLAY
                  340 ;*****************************************************************
                  341
0100              342          ORG     100H
                  343 REFRSH:
0100 23F0         344          MOV     A,#0F0H
0102 3A           345          OUTL    P2,A            ;TURN OFF DIGITS
                  346
                  347          LDA     DISFLG
0103 B825         348+         MOV     R0,#DISFLG
0105 F0           349+         MOV     A,@R0
0106 C60E         350          JZ      NOSWAP          ;CHECK DISPLAY FLAG
                  351          LDA     ANGLE
0108 B823         352+         MOV     R0,#ANGLE
010A F0           353+         MOV     A,@R0
010B 47           354          SWAP    A
010C 2411         355          JMP     CONT2
                  356
                  357 NOSWAP:  LDA     ANGLE
010E B823         358+         MOV     R0,#ANGLE
0110 F0           359+         MOV     A,@R0
0111 530F         360 CONT2:   ANL     A,#0FH          ;MASK UPPER NIBBLE
0113 032D         361          ADD     A,#(LOW TABL)
0115 A3           362          MOVP    A,@A            ;GET CHAR CODE
                  363
                  364          STA     SEGDAT
0116 B824         365+         MOV     R0,#SEGDAT
0118 A0           366+         MOV     @R0,A
0119 39           367          OUTL    P1,A            ;TURN ON THE SEGMENTS
                  368
                  369          LDA     DISFLG
011A B825         370+         MOV     R0,#DISFLG
011C F0           371+         MOV     A,@R0
011D C623         372          JZ      DIG1
                  373
011F 2310         374 DIG2:    MOV     A,#10H
0121 2424         375          JMP     CONT3
                  376
0123 27           377 DIG1:    CLR     A
                  378
0124 3A           379 CONT3:   OUTL    P2,A            ;TURN ON THE DIGIT
                  380
                  381          LDA     DISFLG
0125 B825         382+         MOV     R0,#DISFLG
0127 F0           383+         MOV     A,@R0
0128 37           384          CPL     A
                  385          STA     DISFLG          ;CHANGE DISPLAY FLAG
0129 B825         386+         MOV     R0,#DISFLG
012B A0           387+         MOV     @R0,A
                  388
012C 83           389          RET
                  390
                  391 TABL:
012D 3F           392          DB      00111111B       ; 0
012E 06           393          DB      00000110B       ; 1
012F 5B           394          DB      01011011B       ; 2
0130 4F           395          DB      01001111B       ; 3
0131 66           396          DB      01100110B       ; 4
0132 6D           397          DB      01101101B       ; 5
0133 7D           398          DB      01111101B       ; 6
0134 07           399          DB      00000111B       ; 7
0135 7F           400          DB      01111111B       ; 8
0136 67           401          DB      01100111B       ; 9
                  402
0137              403 HERE1    EQU     $
                  404
                  405 $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    9

 LOC  OBJ              SOURCE STATEMENT

               ;*********************************************************************
               ;
               ;                  LOOKUP   TABLE   FOR THE PHASE ANGLE
               ;
               ;*********************************************************************
               ;
00C9                      ORG     TAB

00C9 00                   DB      00H
00CA 06                   DB      06H
00CB 0C                   DB      0CH
00CC 11                   DB      11H
00CD 17                   DB      17H
00CE 1D                   DB      1DH
00CF 23                   DB      23H
00D0 29                   DB      29H
00D1 2E                   DB      2EH
00D2 34                   DB      34H
00D3 3A                   DB      3AH
00D4 40                   DB      40H
00D5 46                   DB      46H
00D6 4B                   DB      4BH
00D7 51                   DB      51H
00D8 57                   DB      57H
00D9 5D                   DB      5DH
00DA 63                   DB      63H
00DB 68                   DB      68H
00DC 6E                   DB      6EH
00DD 74                   DB      74H
00DE 7A                   DB      7AH
00DF 80                   DB      80H
00E0 85                   DB      85H
00E1 8B                   DB      8BH
00E2 91                   DB      91H
00E3 97                   DB      97H
00E4 97                   DB      97H
00E5 97                   DB      97H
00E6 97                   DB      97H

               $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    10

 LOC  OBJ         SEQ          SOURCE STATEMENT

0137                              ORG     HERE1

                           $       INCLUDE(CONBCD)
                           ;***********************************************************************
                           ;*
                           ;*                  CONBCD
                           ;*
                           ;***********************************************************************
                           ;*
                           ;*                  THIS UTILITY CONVERTS AN 8 BIT BINARY VALUE TO BCD
                           ;*                  AT ENTRY:
                           ;*                          A = 8 BIT BINARY VALUE
                           ;*
                           ;*                  AT EXIT:
                           ;*                          A = BCD VALUE
                           ;***********************************************************************
                           ;
0005                       TEMP1   SET      R5
0006                       TEMP2   SET      R6
                           ;
                           ;1 CONVERT_TO_BCD
                           CONBCD:
                           ;1 BCDACC:=0
0137 BE00                          MOV      TEMP2,#00
                           ;1 COUNT:=8
0139 BB08                          MOV      COUNT,#08
                           ;1 REPEAT
                           BCDCOB:
                           ;2    BIN:=BIN*2
013B 97                            CLR      C
013C F7                            RLC      A
                           ;2    BCD:=BCD*2+CARRY
013D AD                            MOV      TEMP1,A
013E FE                    BCDOC:  MOV      A,TEMP2
013F 7E                            ADDC     A,TEMP2
0140 57                            DA       A
0141 AE                            MOV      TEMP2,A
0142 FD                            MOV      A,TEMP1
                           ;2    IF CARRY FROM BCDACC GOTO ERROR EXIT
0143 F649                          JC       BCDCOD
                           ;2    COUNT:=COUNT-1
                           ;1 UNTIL COUNT=0
0145 EB3B                          DJNZ     COUNT,BCDCOB
0147 97                            CLR      C        ; CLEAR CARRY TO INDICATE NORMAL TERMINATION
0148 FE                            MOV      A,TEMP2  ; PUT BCD VALUE IN ACCUMULATOR FOR RETURN
                           ;1 END CONVERT_TO_BCD
0149 83                    BCDCOD: RET
                           $EJECT
```

ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0   PAGE   11

```
LOC  OBJ         SEQ        SOURCE STATEMENT

                           $      INCLUDE(CONB2)
                           ;*************************************************************************
                           ;*
                           ;*     CONBIN
                           ;*
                           ;*========================================================================
                           ;*
                           ;*     THIS UTILITY CONVERTS A 2 DIGIT BCD VALUE TO BINARY
                           ;*     AT ENTRY:
                           ;*       A = BCD VALUE
                           ;*
                           ;*     AT EXIT:
                           ;*       A = EIGHT BITS OF THE BINARY RESULT
                           ;*
                           ;*************************************************************************
                           ;
                           ;
                           ;
0005                       TEMP1   SET     R5
0006                       TEMP2   SET     R6
0007                       TEMP3   SET     R7
                           ;
                           ;1 CONVERT_TO_BINARY
                           CONBIN:
                           ;1 COUNT:=DIGITPAIR
014A BB01                          MOV     COUNT,#DIGPR
                           ;  REPEAT
                           CONBLP:
                           ;     BIN:=BIN*10
014C AF                            MOV     TEMP3,A
014D 47                            SWAP    A
014E 530F                          ANL     A,#0FH
0150 345A                          CALL    CONB10
                           ;     BIN:=BIN+MEM(R0)[7-4]
0152 AE                            MOV     TEMP2,A
0153 FF                            MOV     A,TEMP3
0154 530F                          ANL     A,#0FH
0156 6E                            ADD     A,TEMP2
                           ;      COUNT:=COUNT-1
                           ;  UNTIL COUNT=0
0157 EB4C                          DJNZ    COUNT,CONBLP
                           ;  END CONVERT_TO_BINARY
0159 83                    CONBER: RET
                           ;
                           $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0              PAGE    12

 LOC  OBJ         SEQ        SOURCE STATEMENT

                = 541  ;
                = 542  ;          UTILITY TO MULTIPLY BIN BY 10
                = 543  ;          CARRY WILL BE SET IF OVERFLOW OCCURS
                = 544  ;
 015A AD        = 545  CONB10: MOV      TEMP1,A ; SAVE A
                = 546
 015B 97        = 547          CLR      C
 015C F7        = 548          RLC      A        ; BIN:=BIN*2
                = 549
 015D F7        = 550          RLC      A        ; BIN:=BIN*4
                = 551
 015E 6D        = 552          ADD      A,TEMP1 ; BIN:=BIN*5
                = 553
 015F F7        = 554          RLC      A        ; BIN:=BIN*10
                = 555
 0160 83        = 556          RET
                = 557
                = 558  ;
                = 559  $EJECT
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER,  V2.0                  PAGE    13

 LOC  OBJ         SEQ        SOURCE STATEMENT

                  560        END


USER SYMBOLS
ANGLE  0023   BAK1   00AB   BAK2   00B5   BCDCOB 013B   BCDCOD 0149
BCDOC  013E   BCDTR  0021   BINTR  0020   CLRFLG 004C   CNBCD  0137
CNTDWN 00C1   CONB10 015A   CONBER 0159   CONBIN 014A   CONBLP 014C
CONT2  0111   CONT3  0124   CONTON 0058   COUNT  0003   DECREM 0049
DELTA  0026   DIG1   0123   DIG2   011F   DIGPR  0001   DISFLG 0025
DLGT10 0007   FIRE   00C5   HERE1  0137   HOLD   0054   ICNT   0004
INCREM 003B   INIT   0000   INITZ  0098   INWAIT 0063   IPWAIT 0016
LDA    0000   LOOK   004F   NOSWAP 010E   OFFTIM 0003   OTCALC 0038
PASTOR 0022   PHLTSV 0006   PHSANG 0005   PWRON  0002   REFRSH 0100
SEGDAT 0024   SELRB0 00B1   SELRB1 00A7   SETVAL 0004   STA    0001
START  000E   TAB    00C9   TABL   012D   TEMP1  0005   TEMP2  0006
TEMP3  0007   TIME   0002   TRIAC  00BB   VOWAIT 0006   WAITST 000A
XA     0002

ASSEMBLY COMPLETE,  NO ERRORS
```

```
ISIS-II ASSEMBLER SYMBOL CROSS REFERENCE, V2.0                    PAGE    1

ANGLE    210    235#   352    358
BAK1     288#   291
BAK2     300#   303
BCDCOB   475#   490
BCDCOD   487    494#
BCDOC    481#
BCDTR    189    195    201    207    233#
BINTR    158    164    170    176    232#
CLRFLG    97     99    104    106    112#
CNBCD    198    469#
CNTDWN   326#   327
CONB10   528    545#
CONBER   538#
CONBIN   167    519#
CONBLP   523#   536
CONT2    355    360#
CONT3    375    379#
CONTON   131    134#
COUNT    252#   473    490    521    536
DECREM    93    108#
DELTA     73     81    238#
DIG1     372    377#
DIG2     374#
DIGPR    254#   521
DISFLG   237#   268    348    370    382    386
DLGT10    86    103    114    245#
FIRE     326    329#
HERE1    403#   448
HOLD     131#   132
ICNT     253#
INCREM    95#
INIT      17#
INITZ     17    226#
INWAIT   144#   146
IPWAIT    48#    50
LDA        7#    80    138    163    175    194    206    347    351    357    369    381
LOOK      70    122#
NOSWAP   350    357#
OFFTIM    95    101    105    108    110    241#   321
OTCALC    77     85     92#
PASTOR   234#
PHLTSV    78     92    113    244#
PHSANG    68    125    187    243#
PWRON     24#    25
REFRSH    41    137    343#
SEGDAT   139    236#   365
SELRB0   173    204    297#
SELRB1   161    192    285#
SETVAL    65    178    242#
STA       12#    72    157    169    188    200    209    267    364    385
START     30     37#
TAB      122    334#   413
TABL     361    391#
TEMP1    465#   480    485    514#   545    552
TEMP2    466#   471    481    482    484    492    515#   530    533
TEMP3    516#   525    531
TIME      53    123    240#
TRIAC     55    148    316#
VOWAIT    24     27#    28
WAITST    27     30#    31    214
XA       251#

CROSS REFERENCE COMPLETE
```

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080

Printed in USA/TP 44/0480/30K SC HG